

INSTRUCTION MANUAL

NTB — 500A

(FOR CONTROL FUNCTION)

Thank you for purchasing KYOWA's product NTB-500A Medium Speed Network Terminal Box.

Read this Instruction Manual carefully in order to make full use of the high performance capabilities of the product.

Do not use the product in methods other than described in this Manual.

This Instruction Manual describes the Control Function of the NTB-500A.

For hardware, refer to the “Instruction Manual of the NTB-500A Medium Speed Network Terminal Box (For Hardware).”

For software, refer to the “Instruction Manual of the NTB-500A Medium Speed Network Terminal Box (For Software).”

Windows XP/Vista/7 are trademarks of Microsoft Corporation, USA.

Generally, company names and trade names described in this Instruction Manual are trademarks or registered trademarks of the companies.

This Instruction Manual may not be copied or reproduced, in whole or part, without consent of KYOWA.

Copyright © Kyowa Electronic Instruments Co., Ltd. All rights reserved.

The contents of the Instruction Manual are subjected to change without prior notice.

CONTENTS

1. OUTLINE	1
2. CONTROL FUNCTIONS	2
2-1 RETURN VALUES OF THE CONTROL FUNCTIONS (COMMON)	3
2-2 ARGUMENT OF CONTROL FUNCTIONS (COMMON)	3
2-3 DETAILS OF THE CONTROL FUNCTIONS.....	5
2-3-1 NtbInitializeLib.....	5
2-3-2 NtbTerminationLib	5
2-3-3 NtbGetCanDeviceInfo	5
2-3-4 NtbOpenCanDevice.....	6
2-3-5 NtbCloseCanDevices.....	6
2-3-6 NtbCheckConnect	7
2-3-7 NtbSetName.....	7
2-3-8 NtbSetChMode	8
2-3-9 NtbGetChMode.....	9
2-3-10 NtbSetMeasChStatus	9
2-3-11 NtbGetMeasChStatus	10
2-3-12 NtbSetSampling.....	10
2-3-13 NtbGetSampling.....	11
2-3-14 NtbCreateMeasInfo	11
2-3-15 NtbStartMoni.....	11
2-3-16 NtbStopMoni.....	12
2-3-17 NtbGetMoniData.....	12
2-3-18 NtbGetRelationShip.....	13
2-3-19 NtbSensorCheck.....	13
2-3-20 NtbStartSnsChkZeroMeas	14
2-3-21 NtbSetZeroMeasStatus	15
2-3-22 NtbGetZeroMeasStatus.....	15
2-3-23 NtbSetZeroCorrStatus	15
2-3-24 NtbGetZeroCorrStatus	16
2-3-25 NtbSetZeroValue	16
2-3-26 NtbGetZeroValue	16
2-3-27 NtbSetReferenceJunction	17
2-3-28 NtbGetReferenceJunction	17
2-3-29 NtbGetTerminalTemperature.....	17
2-3-30 NtbGetTedsCalibrationFactor	18
2-3-31 NtbDeleatAllCanDeviceBuff.....	18
2-3-32 NtbCheckCanDeviceBps	18
2-3-33 NtbReSet.....	18
3. SIMPLIFIED SOFTWARE.....	19
3-1 SIMPLIFIED SOFTWARE SPECIFICATIONS.....	19
3-2 HOW TO START UP THE SIMPLIFIED SOFTWARE	21

1. OUTLINE

This Instruction Manual describes control functions to create the original control software for the NTB-500A Medium Speed Network Terminal Box (hereinafter referred to as the NTB-500A).

Be sure to create the control software on the computer that have either Windows XP/Vista/7. All control functions are stored in the Dynamic Link Library (DLL) for the NTB-500A.

For details, refer to commercially available textbooks, etc.

The DLL for the NTB-500A communicates the NTB-500A by using the USB-CAN converter (Kvaser Inc., model: Leaf Light HS). You can't use multiple USB-CAN converters. You cannot use the other USB-CAN converters. To use the USB-CAN converter (Leaf Light HS), install the device driver of the USB-CAN converter. For installation, refer to the "Instruction Manual of the NTB-500A Medium Speed Network Terminal Box (For Hardware)."

OPERATION ENVIRONMENT

OS	Microsoft Windows XP
	Microsoft Windows Vista
	Microsoft Windows 7
Memory	1GByte (Microsoft Windows XP)
	2GByte (Microsoft Windows Vista / Microsoft Windows 7)

Necessary Files

① canli32.dll	CAN communication library
② kcanl.sys	The device driver of the USB-CAN converter
③ NtbBaseLib.dll	The DLL for controlling the NTB-500A The NtbBaseLib.dll is a file of code containing control functions. The NtbBaseLib.dll has 32-bit version and 64-bit version. Use the DLL that matches the system requirements.

2. CONTROL FUNCTIONS

List of Control Function

No.	Function Name	Function
1	NtbInitializeLib	Initialize the NtbBaseLib.dll.
2	NtbTerminationLib	End the NtbBaseLib.dll.
3	NtbGetCanDeviceInfo	Get the information of the USB-CAN converter.
4	NtbOpenCanDevice	Connect the USB-CAN converter.
5	NtbCloseCanDevices	End the connection of the USB-CAN converter.
6	NtbCheckConnect	Communication checking
7	NtbSetName	Set the device name.
8	NtbSetChMode	Set the CH mode.
9	NtbGetChMode	Load the CH mode.
10	NtbSetMeasChStatus	Set the channels to be measured.
11	NtbGetMeasChStatus	Load the channels to be measured.
12	NtbSetSampling	Set the sampling frequency.
13	NtbGetSampling	Load the sampling frequency.
14	NtbCreateMeasInfo	Pass the information, required for measurements, to the NtbBaseLib.dll.
15	NtbStartMoni	Start measuring data
16	NtbStopMoni	Stop measuring data.
17	NtbGetMoniData	Get the measured data.
18	NtbGetRelationShip	Load whether the NTB-500A is master or slave.
19	NtbSensorCheck	Sensor checking
20	NtbStartSnsChkZeroMeas	ZERO balance and sensor checking
21	NtbSetZeroMeasStatus	Set the ZERO-balance channels
22	NtbGetZeroMeasStatus	Load the ZERO-balance channels
23	NtbSetZeroCorrStatus	Set the ZERO-value-subtraction channels
24	NtbGetZeroCorrStatus	Load the ZERO-value-subtraction channels
25	NtbSetZeroValue	Set the ZERO value.
26	NtbGetZeroValue	Load the ZERO value.
27	NtbSetReferenceJunction	Set the reference junction compensation.
28	NtbGetReferenceJunction	Load the settings of the reference junction compensation.
29	NtbGetTerminalTemperature	Load the temperature of the terminal.
30	NtbGetTedsCalibrationFactor	Calculate the calibration constant by loading the TEDS information.
31	NtbDeleatAllCanDeviceBuff	Clear buffers of all the USB-CAN converters.
32	NtbCheckCanDeviceBps	Load the baud rate automatically.
33	NtbReSet	Initialize the NTB-500A.

2-1 RETURN VALUES OF THE CONTROL FUNCTIONS (COMMON)

Unless stated, the return values are as follows.

- 0 Normal end
- 1 Command error or parameter error
- 2 NTB-500A hardware configuration error
- 3 USB-CAN converter not connected.
- 4 USB-CAN converter error
- 5 Could not connect the USB-CAN converter
- 6 DLL is not initialized yet. (not calling the NtbInitializeLib.)
- 7 The USB-CAN converter is not in the available state. (not calling the NtbOpenCanDevices)
- 8 Cannot load the canlib32.dll.
- 9 No device driver of the USB-CAN converter
- 10 The number of the USB-CAN converters, connected to the PC, exceeded the maximum value.
- 11 CAN communication frame error
- 12 No message on the USB-CAN converter buffer
- 13 Send error
- 14 Receive error
- 15 Timeout error while receiving data
- 16 The number of the NTB-500A units, connected to the PC, exceeded the maximum value.
- 17 The NTB-500A units, having the same BOX No., exist.
- 18 The specified BOX No. does not exist.
- 19 A buffer overflow error occurs on the NTB-500A or on the USB-CAN converter.
- 20 No command available
- 21 The NTB-500A is not connected.

2-2 ARGUMENT OF CONTROL FUNCTIONS (COMMON)

To specify the NTB-500A, use the boxno.

The boxno is the BOX No., pre-set BOX No. switch, of the NTB-500A.

Available values are 0 to 99. You can operate the NTB-500A units that do not have sequential numbers (sequential boxno).

However, you cannot communicate correctly when multiple NTB-500A units have the same BOX No.

Be sure not to set the same BOX No.

You can control the NTB-500A units up to 8 units.

With the NTB-500A, use the following functions.

For setting parameters: NtbSet***

For loading parameters: NtbGet***

To load the information from the connected NTB-500A units (up to 8 units), pass "0" as the boxno of the NtbGet***.

The information is stored in numerical order of the BOX No.

Suppose the 3 NTB-500A units are connected, the BOX No. are 1, 3, and 5, and execute the NtbGetChMode (after-mentioned) with the boxno=0.

mode[0]: CH1 mode of the BOX No. 1

mode[8]: CH1 mode of the BOX No. 3

mode[16]: CH1 mode of the BOX No. 5

Do not pass "0" as the boxno of the NtbSet***. The NTB-500A will return an error.

2-3 DETAILS OF THE CONTROL FUNCTIONS

This section describes the details of the control functions.

2-3-1 NtbInitializeLib

Function	Initialize the NtbBaseLib.dll. Call the function first. Or, you cannot call any function.
Definition	int __stdcall NtbInitializeLib(void)
Argument	None
Others	Call the NtbInitializeLib to start up the application program. Call the NtbTerminationLib to end the application program.

2-3-2 NtbTerminationLib

Function	End the NtbBaseLib.dll. Call the function to end the application program.
Definition	int __stdcall NtbTerminationLib(void)
Argument	None
Others	Call the function to end the application program. Or, it will cause Windows to become unstable.

2-3-3 NtbGetCanDeviceInfo

Function	Get the information of the USB-CAN converter (Leaf Light HS).
Definition	int __stdcall NtbGetCanDeviceInfo(char *num, char *id, int *serial)
Argument	num The number of the USB-CAN converter, connected to the PC. id ID of the USB-CAN converter Pass the char [4]. Serial Serial number of the USB-CAN converter Pass the int [4].
Others	To connect the USB-CAN converter, pass the ID of the USB-CAN converter to be used as the argument of the NtbOpenCanDevices.

2-3-4 NtbOpenCanDevice

Function	Connect the USB-CAN converter having the ID, passed by the argument. Before communicating, be sure to call the function to connect the USB-CAN converter.
Definition	int __stdcall NtbOpenCanDevices(char num, char *id, long *bps, unsigned char ntbttype, char flag)
Argument	num Pass "1" id Pass the ID of the USB-CAN converter to be connected. bps Pass the baud rate of the USB-CAN converter to be connected. See the following. 1000000 (1Mbps) 500000 (500kbps) 250000 (250kbps) ntbttype Pass "1". flag Pass "0".
Others	Set the bps that matches the cable length of the NTB-500A. Cable length of the NTB-500A is 20 m: 1Mbps Cable length of the NTB-500A is 80 m: 500kbps Cable length of the NTB-500A is 100 m: 250kbps

Be sure to set the bps that matches the cable length of the NTB-500A.
Or, the communication can not be established.

2-3-5 NtbCloseCanDevices

Function	Disconnect the USB-CAN converter.
Definition	int __stdcall NtbCloseCanDevices(char num, char *id)
Argument	num Pass "1" id id of the USB-CAN converter to be disconnected.
Others	Call the function to end the communication between the NTB-500A and USB-CAN converter.

2-3-6 NtbCheckConnect

Function	Execute the communication checking of the NTB-500A. After the connection succeeded, the standard information is loaded.
Definition	int __stdcall NtbCheckConnect(char *num, char *boxno, unsigned char *ntbtype, unsigned char *unit, int *sampling)
Argument	num The number of the NTB-500A units, connected to the PC You can control the NTB-500A units up to 8 units. boxno The BOX No. of the NTB-500A units, connected to the PC Pass the char [8]. ntbtype Model of the NTB, connected to the PC unsigned char [8] When connecting the NTB-500A, ntbtype is 80. unit Measuring unit's information of the NTB-500A Every NTB-500A can mount up to 2 measuring units. unsigned char [8 * 2] Sampling Sampling frequency of the NTB-500A, connected to the PC
Others	Call the function to connect the NTB-500A. After changing the hardware configuration (measuring unit, BOX No., the number of NTB-500A units) of the NTB-500A, check the connection by using the function.

2-3-7 NtbSetName

Function	Set the device name of one NTB-500A.
Definition	int __stdcall NtbSetChMode(char boxno, char *name)
Argument	boxno Pass the BOX No. of the NTB-500A to be set. boxno=0: Error name Pass the device name to be set. char [32] This is a 32-byte string, including NULL bytes.

2-3-8 NtbSetChMode

Function: Set the CH mode of one NTB-500A.

Definition **int __stdcall NtbSetChMode(char boxno, unsigned char *mode)**

Argument **boxno** Pass the BOX No. of the NTB-500A to be set.

boxno=0: Error

mode Pass the CH mode to be set.

unsigned char [8]

mode[0]: CH1, mode[7]: CH8

See the following.

No measuring unit

0: SKIP

CH mode of the NTB-50A

20: Quarter bridge system, 120 ohm, 2-wire	$30\text{ k} \times 10^{-6}$ range	(Low range)
21: Quarter bridge system, 120 ohm, 2-wire	$300\text{ k} \times 10^{-6}$ range	(High range)
22: Quarter bridge system, 120 ohm, 3-wire	$30\text{ k} \times 10^{-6}$ range	(Low range)
23: Quarter bridge system, 120 ohm, 3-wire	$300\text{ k} \times 10^{-6}$ range	(High range)
24: Half bridge system, $30\text{ k} \times 10^{-6}$ range		(Low range)
25: Half bridge system, $300\text{ k} \times 10^{-6}$ range		(High range)
26: Full bridge system, $30\text{ k} \times 10^{-6}$ range		(Low range)
27: Full bridge system, $300\text{ k} \times 10^{-6}$ range		(High range)

CH mode of the NTB-51A

30: Voltage	10 V range (Low range)
31: Voltage	50V range (High range)
32: Thermocouple	TC/K
33: Thermocouple	TC/T

Others

CH modes vary with the measuring unit.

When no measuring unit exists, set the CH "0".

2-3-9 NtbGetChMode

Function	Load the CH mode from the NTB-500A.
Definition	int __stdcall NtbGetChMode(char boxno, unsigned char *mode)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load the CH modes for the number of NTB-500A units, pass "0" as the boxno. mode CH mode unsigned char [8 * the number of NTB-500A units]
Others	When executing the boxno=0, pay attention to the number of elements of the mode. When connecting 2 NTB-500A units, the number of elements of mode is 16 (8 channels * 2 units). mode[0]: The CH1 mode of the BOX No.01 mode[8]: The CH1 mode of the BOX No.02

2-3-10 NtbSetMeasChStatus

Function	Set the channels to be measured of one NTB-500A.
Definition	int __stdcall NtbSetMeasChStatus(char boxno, unsigned char *sts)
Argument	boxno BOX No. of the NTB-500A to be set sts Pass the status of the channels to be measured unsigned char [8] 0: Measurement OFF, 1: Measurement ON, 2: No channel to be measured (no measuring unit)
Others	boxno=0: Parameter error When no measuring unit exists, set the sts "2". With the NTB-500A, the maximum number of channels to be measured varies with the sampling frequency and cable length of the NTB-500A. See the following.

Table1 The maximum number of channels to be measured, the sampling frequency and cable length of the NTB-500A

Sampling frequency (Hz)	The maximum number of channels to be measured		
	Cable length 20 m	Cable length 80 m	Cable length 100 m
1 k	8	4	--
500	16	8	4
200	40	20	8
100	64	40	20
50	64	64	40
20 to 1	64	64	64

NOTE

- Do not set the number of channels to be measured to a value greater than the maximum.
Or, you cannot measure data correctly.

2-3-11 NtbGetMeasChStatus

Function	Load the channels to be measured from the NTB-500A.
Definition	int __stdcall NtbSetMeasChStatus(char boxno, unsigned char *sts)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load the status of the channels to be measured for the number of NTB-500A units, pass "0" as the boxno. sts Status of the channels to be measured unsigned char [8 * the number of NTB-500A units]
Others	When connecting 2 NTB-500A units, the number of elements of sts is 16 (8 channels * 2 units). sts[0]: The status of the CH1 of the BOX No.0 sts[8]: The status of the CH1 of the BOX No.2 For the maximum number of channels to be measured, the sampling frequency and cable length of the NTB-500A, see Table 1.

2-3-12 NtbSetSampling

Function	Set the sampling frequency of one NTB-500A.
Definition	int __stdcall NtbSetSampling(char boxno, unsigned int sampling)
Argument	boxno BOX No. of the NTB-500A to be set boxno=0 (recommended) sampling Sampling frequency to be set 1000: Sampling frequency 1kHz 500: Sampling frequency 500 Hz 200: Sampling frequency 200Hz 100: Sampling frequency 100Hz 50: Sampling frequency 50Hz 20: Sampling frequency 20Hz 10: Sampling frequency 10Hz 5: Sampling frequency 5Hz 2: Sampling frequency 2Hz 1: Sampling frequency 1Hz
Others	Set the same sampling frequency to all NTB-500A units.

2-3-13 NtbGetSampling

Function	Load the sampling frequency of the NTB-500A.
Definition	int __stdcall NtbGetSampling(char boxno, unsigned int *sampling)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load the sampling frequency for the number of NTB-500A units, pass "0" as the boxno. Sampling Sampling frequency int [the number of NTB-500A units]
Others	When connecting 2 NTB-500A units, the number of elements of sampling is 2. sampling[0]: The sampling frequency of the BOX No. 01 sampling[1]: The sampling frequency of the BOX No. 02

2-3-14 NtbCreateMeasInfo

Function	Pass the information, required for measurements, to the DLL.
Definition	int __stdcall NtbCreateMeasInfo(char num, char *boxno, unsigned char *chmode, unsigned char *measch)
Argument	num Pass the number of the NTB-500A units, connected to the PC (1 to 8) boxno Pass the BOX No. of the NTB-500A units, connected to the PC unsigned char [the number of NTB-500A units] chmode Pass the CH mode. The parameters are the same as the NtbSetChMode. char [8 * the number of NTB-500A units] measch Pass the channels to be measured. The parameters are the same as the NtbSetMeasChStatus. unsigned char [8 * the number of NTB-500A units]
Others	When connecting 2 NTB-500A units, the number of elements of chmode and measch are 16 (8 channels * 2 units). chmode[0] to chmode [7]: CH modes of the BOX No. boxno[0] chmode[8] to chmode [15]: CH modes of the BOX No. boxno[1] Set the measch in the same manner as the chmode.

2-3-15 NtbStartMoni

Function	Start measuring data.
Definition	int __stdcall NtbStartMoni(void)
Argument	None
Others	After starting measurements, periodically get data by calling the NtbGetMoniData Call the function to stop measurements. After starting measurements, do not call functions other than the NtbGetMoniData and NtbStopMoni. Or, an error occurs.

2-3-16 NtbStopMoni

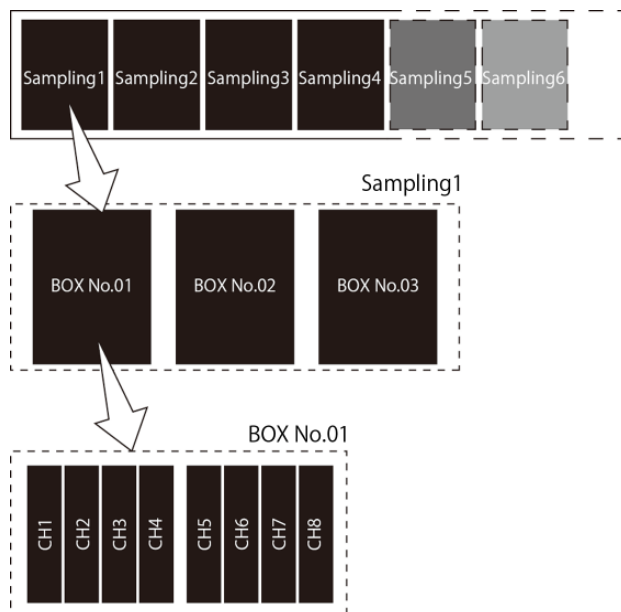
Function	Stop measuring data.
Definition	int __stdcall NtbStopMoni(void)
Argument	None
Others	Call the function to stop measurements.

2-3-17 NtbGetMoniData

Function	Get the measured data.
Definition	int __stdcall NtbGetMoniData(float *buff, long bufsize, int *err boxno)
Argument	buff The measured data float bufsize Pass the data size (the number of bytes) of the buff. err boxno BOX No. of the NTB-500A where error occurred
Return value	Return value ≥ 0 : The return value is the number of measurements. Return value < 0 : Error
Others	Call the NtbGetMoniData by every 100 msec to get data. Set the number of elements, pass as the buff, 8000 or more.

(8000 = Sampling frequency (1kHz) * The number of channels to be measured (8 channels) * The number of data for 1 second)

See the following.



2-3-18 NtbGetRelationship

Function	Load whether the NTB-500A is master or slave.
Definition	int __stdcall NtbGetRelationship(char boxno ,unsigned char *relation)
Argument	boxno BOX No. of the NTB-500A to be loaded To load settings of all the NTB-500A units, set the boxno=0. relation Master or slave 0: Master, 1: Slave unsigned char [8 * the number of NTB-500A units]
Others	When connecting multiple master NTB-500A units, settings (master/slave) are stored in the relation in the numerical order of the BOX No. When multiple master units exist, correct the settings by using the NtbSetRelationship.

2-3-19 NtbSensorCheck

Function	Execute sensor checking whether a sensor (strain and thermocouple) is working correctly.
Definition	int __stdcall NtbSensorCheck(char boxno, unsigned char* result)
Argument	boxno Pass the BOX No. of the NTB-500A to be checking. To execute sensor checking of all the NTB-500A units, set the boxno=0. result Results of sensor checking unsigned char [8 * the number of NTB-500A units] See the following. 0: Error channel 1: Correct channel 2: Not applicable
Others	When the status of the channels to be measured is 0 (OFF), the sensor checking is executed.

2-3-20 NtbStartSnsChkZeroMeas

Function Measure the ZERO balance after executing sensor checking.

With the NTB-500A, measuring the ZERO balance is called the "ZERO balance" and the measured ZERO balance value is called the "ZERO value".

Subtracting the ZERO value from the measured value is called the "ZERO value subtraction".

Measure the ZERO balance: ZERO balance

The measured ZERO balance value: ZERO value

Subtract the ZERO value from the measure value: ZERO value subtraction

The ZERO value is saved in the measuring unit of the NTB-500A.

Definition **int __stdcall NtbStartSnsChkZeroMeas(char boxno, float *pfVal, unsigned char *pbResult)**

Argument **boxno** Pass the BOX No. of the NTB-500A.
To execute sensor checking and ZERO balance of all the NTB-500A units, set the boxno=0.
pfVal ZERO value float [16 * the number of NTB-500A units]
Every ZERO value has low-range ZERO value and high-range ZERO value. The pfVal has 2 ZERO values per one channel.
When no measuring unit exists, the ZERO value is returned.
The data values are low-range ZERO values ×8 channels → high-range ZERO values ×8 channels.

Example:

When connecting 2 NTB-500A units (BOX No.1 and BOX No.2)
pfVal[0]: Low-range ZERO value of the CH1, BOX No.1
pfVal[8]: High-range ZERO value of the CH1, BOX No.1
pfVal[16]: Low-range ZERO value of the CH1, BOX No.2
pfVal[24]: High-range ZERO value of the CH1, BOX No.2

When there is one range, such as a thermocouple, the ZERO value is saved in the low-range ZERO value. The high-range ZERO value is 999,999.

When no measuring unit exists, the low-range ZERO value is 999,999 and high-range ZERO value is 999,999.

pdResult

Results of sensor checking The parameters are the same as the NtbSensorCheck.

Others The function is available when the A/D conversion is being stopped.
The ZERO value, in the pfval, varies with the ZERO balance ON/OFF and ZERO value subtraction ON/OFF.

See the following.

ZERO balance	ZERO value subtraction	ZERO value	Renewing the ZERO value
ON	ON	When the ZERO value is within the range: ZERO value	Yes
OFF		When the ZERO value exceeds the range: 999,999	None
ON	OFF	ZERO value	Yes
OFF			None

2-3-21 NtbSetZeroMeasStatus

Function	Set the ZERO balance ON/OFF of one NTB-500A. ON: Measure the ZERO value when executing the ZERO balance. OFF: Do not measure the ZERO value when executing the ZERO balance.
Definition	int __stdcall NtbSetZeroMeasStatus(char boxno, unsigned char *flag)
Argument	boxno Pass the BOX No. of the NTB-500A to be set. boxno=0: Error flag Pass the ZERO balance ON/OFF. unsigned char [8] See the following. 0: ZERO balance OFF, 1: ZERO balance ON
Others	When no measuring unit exists, set the flag "0".

2-3-22 NtbGetZeroMeasStatus

Function	Load the ZERO balance ON/OFF from the NTB-500A.
Definition	int __stdcall NtbGetZeroMeasStatus(char boxno, unsigned char *flag)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load settings of all the NTB-500A units, set the boxno=0. flag ZERO balance ON/OFF unsigned char [8 * the number of NTB-500A units] See the following. 0: ZERO balance OFF, 1: ZERO balance ON
Others	When no measuring unit exists, set the flag "0".

2-3-23 NtbSetZeroCorrStatus

Function	Set the ZERO value subtraction ON/OFF of one NTB-500A.
Definition	int __stdcall NtbSetZeroCorrStatus(char boxno, unsigned char *flag)
Argument	boxno Pass the BOX No. of the NTB-500A to be set. boxno=0: Error flag Pass the ZERO value subtraction ON/OFF. unsigned char [8] See the following. 0: ZERO value subtraction OFF, 1: ZERO value subtraction ON
Others	When the ZERO value subtraction is ON, the NTB-500A usually output a value subtracting the ZERO value from the measured value. However when the ZERO value exceeds the range, the ZERO value and measured value are 999,999.

2-3-24 NtbGetZeroCorrStatus

Function	Load the ZERO value subtraction ON/OFF from the NTB-500A.
Definition	int __stdcall NtbGetZeroCorrStatus(char boxno, unsigned char *flag)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load settings of all the NTB-500A units, set the boxno=0. unsigned char [8 * the number of NTB-500A units] 0: ZERO value subtraction OFF, 1: ZERO value subtraction ON
Others	When connecting 2 NTB-500A units, the number of elements of flag is 16 (8 channels * 2 units).

2-3-25 NtbSetZeroValue

Function	Set the ZERO value of one NTB-500A.
Definition	int __stdcall NtbSetZeroValue(char boxno, float *pfVal)
Argument	boxno Pass the BOX No. of the NTB-500A to be set. boxno=0: Error pfVal Pass the ZERO value. Set the low-range ZERO value and high-range ZERO value. float [16] The data values are low-range ZERO values ×8 channels → high-range ZERO values ×8 channels. Example: When connecting one NTB-500A (BOX No.1) pfVal[0]: Low-range ZERO value of the CH1, BOX No.1 pfVal[8]: High-range ZERO value of the CH1, BOX No.1
Others	When no measuring unit exists, pass the "999,999" as the ZERO value.

2-3-26 NtbGetZeroValue

Function	Load the ZERO value, saved in the NTB-500A.
Definition	int __stdcall NtbGetZeroValue(char boxno, float *pfVal)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load settings of all the NTB-500A units, set the boxno=0. pfVal ZERO value float [16 * the number of NTB-500A units] The data values are low-range ZERO values ×8 channels → high-range ZERO values ×8 channels. Example: When connecting 2 NTB-500A units (BOX No.1 and BOX No.2) pfVal[0]: Low-range ZERO value of the CH1, BOX No.1 pfVal[8]: High-range ZERO value of the CH1, BOX No.1 pfVal[16]: Low-range ZERO value of the CH1, BOX No.2 pfVal[24]: High-range ZERO value of the CH1, BOX No.2

Others	When no measuring unit exists, the ZERO value is 999,999. When there is one range, such as a thermocouple, the ZERO value is saved in the low-range ZERO value. The high-range ZERO value is 999,999. When no measuring unit exists, the low-range ZERO value is 999,999 and high-range ZERO value is 999,999.
--------	--

2-3-27 NtbSetReferenceJunction

Function	Set the reference junction compensation.
Definition	int __stdcall NtbSetReferenceJunction(char boxno, unsigned char ref)
Argument	boxno Pass the BOX No. of the NTB-500A to be set. ref Pass the internal/external. 0: Internal, 1: External
Others	To set the same setting of all the NTB-500A units, set the boxno=0.

2-3-28 NtbGetReferenceJunction

Function	Load the settings of the reference junction compensation.
Definition	int __stdcall NtbGetReferenceJunction(char boxno, unsigned char *ref)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load settings of all the NTB-500A units, set the boxno=0. ref Settings of the reference junction compensation unsigned char [the number of NTB-500A units] 0: Internal, 1: External

2-3-29 NtbGetTerminalTemperature

Function	Load the terminal's temperature of the NTB-51A (voltage thermocouple unit), in the NTB-500A.
Definition	int __stdcall NtbGetTerminalTemperature(char boxno, float *temperature)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. To load settings of all the NTB-500A units, set the boxno=0. temperature Terminal of the temperature float [2 * the number of NTB-500A units] When the number of NTB-500A units is 1 (NTB-50A * 2), the temperatures of terminals are as follows. temperature[0]: Measuring unit 1 temperature[1]: Measuring unit 2
Others	One temperature per one NTB-51A

2-3-30 NtbGetTedsCalibrationFactor

Function	Calculate the calibration constant by loading the TEDS information.
Definition	int __stdcall NtbGetTedsCalibrationFactor(char boxno, double* cal)
Argument	boxno Pass the BOX No. of the NTB-500A to be loaded. boxno=0: Error cal Calibration constant double [8]
Others	When connecting no TEDS sensor, 1. You can use the function when the TEDS template ID is 33.

2-3-31 NtbDeleatAllCanDeviceBuff

Function	Clear buffers of all the USB-CAN converters.
Definition	int __stdcall NtbDeleatAllCanDeviceBuff()
Argument	None

2-3-32 NtbCheckCanDeviceBps

Function	Check the pre-set cable length of the NTB-500A. Load the baud rate of the USB-CAN converter automatically.
Definition	int __stdcall NtbCheckCanDeviceBps(unsigned short *baudrate)
Argument	baudrate
Others	The baud rate, passed as the argument, is used as the initial value for checking. Set the baud rate that matches the NTB-500A to complete checking easily. When the baud rate is unknown, pass "1Mbps". When connecting multiple NTB-500A units with different cable lengths, the communication can not be established. At this time, the baudrate is 0.

2-3-33 NtbReSet

Function	Initialize teh NTB-500A.
Definition	int __stdcall NtbReSet()
Argument	None
Others	See the following.

Measuring Unit	Measurement ON/OFF	Mode	Range	ZERO balance	ZERO value subtraction
NTB-50A (strain unit)	ON	4G	30 k	ON	ON
NTB-51A (voltage/thermocouple unit)	ON	V	10V	OFF	OFF

The ZERO value is "0".

You can initialize the NTB-500A before calling the NtbCheckConnect.

3. SIMPLIFIED SOFTWARE

Simplified software controls the NTB-500A from the PC and displays measured data on the Microsoft Office Excel spreadsheet. You can create the original software by referring to simplified software.

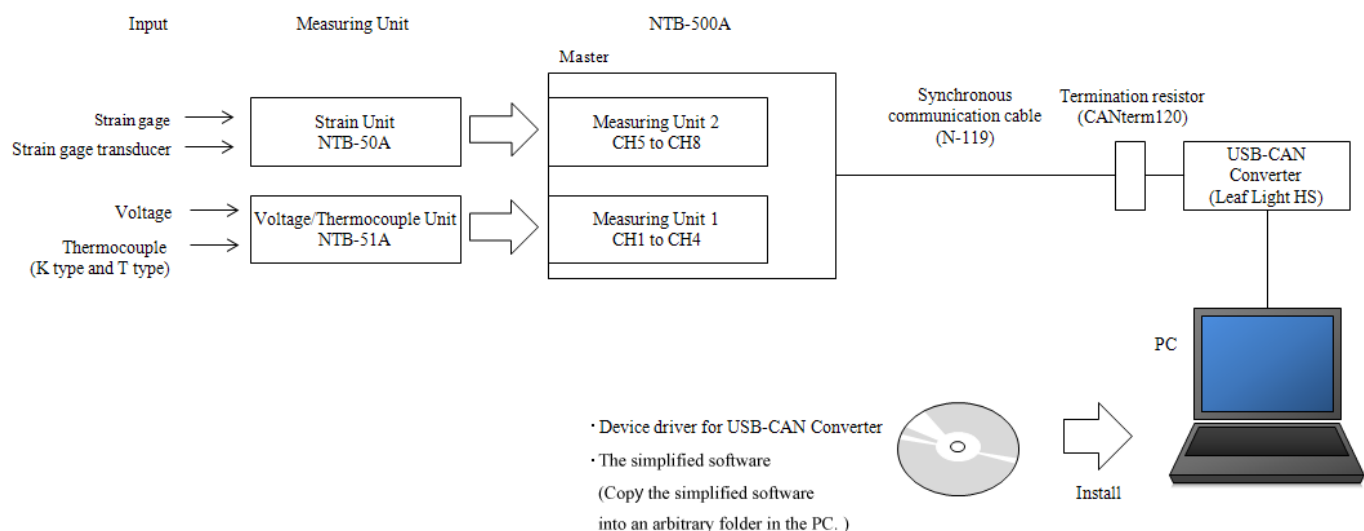
3-1 SIMPLIFIED SOFTWARE SPECIFICATIONS

The simplified software specifications are described as follows.

■ System Configuration

■ PC	
CPU	PentiumIII 1 GHz or more (Pentium4 2 GHz or more recommended)
OS	Microsoft Windows XP (32-bit version)Japanese and English versions Microsoft Windows Vista (32-bit/64-bit version) Japanese and English versions Microsoft Windows 7 (32-bit/64-bit version)Japanese and English versions
Memory	When using the Microsoft Windows XP 1 GByte or more When using the Microsoft Windows Vista } 2 GByte or more When using the Microsoft Windows 7 }
Display	1024 × 768 dots or more
Application	Microsoft Office Excel 2007 (32-bit version) Microsoft Office Excel 2010 (32-bit/64-bit versions)

■ Peripherals	
Measuring Unit	NTB-50A (strain unit) and NTB-51A (voltage/thermocouple unit)
Synchronous communication cable	N-119 (1 m) Note: When requiring the cable other than the above, contact KYOWA or our representatives.
Termination resistor	Vector Japan Co., Ltd. Model: CANterm120
USB-CAN converter	Kvaser Inc. Model: Leaf Light HS



The simplified software is written in Microsoft Visual Basic for Application (VBA).

Use the Microsoft Office Excel 2007 (32-bit version) or Microsoft Office Excel 2010 (32-bit/64-bit version).

■ Software Specifications

The number of NTB-500A units 1 (the maximum number of channels: 8)

The number of measurements: Up to 30000

Measuring conditions

Sampling frequency 1 to 1kHz

With the simplified software, the status of the channels to be measured is always 1 (ON).

Cable length: 20 m

See the following.

Measuring Unit	Measurement ON/OFF	Mode	Range	ZERO balance	ZERO value subtraction
NTB-50A (strain unit)	ON	4G	30k	OFF	OFF
NTB-51A (voltage/thermocouple unit)	ON	V	10V	OFF	OFF

3-2 HOW TO START UP THE SIMPLIFIED SOFTWARE

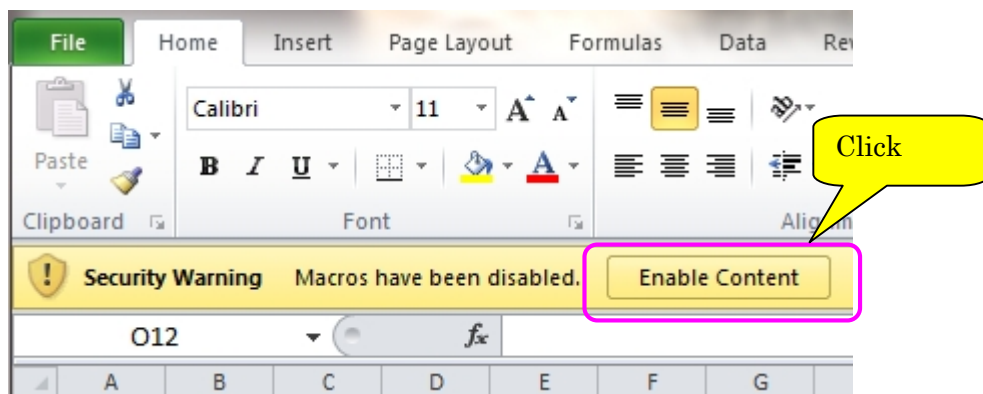
This section describes how to start up the simplified software "NTB500A_VBASample_EN.xlsm".

Copy the "NTB500A_VBASample_EN.xlsm" and "NtbBaseLib.dll", stored in the accessory CD, into an arbitrary folder in the PC. Store the above 2 files in the same folder.

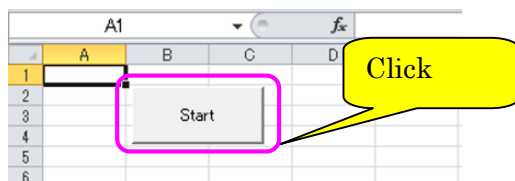
(1) Start up the "NTB500A_VBASample_EN.xlsm".

A security warning will appear. Select Enable Content in the Security Warning at the top of the spreadsheet.

Reference image: Microsoft Office Excel 2010

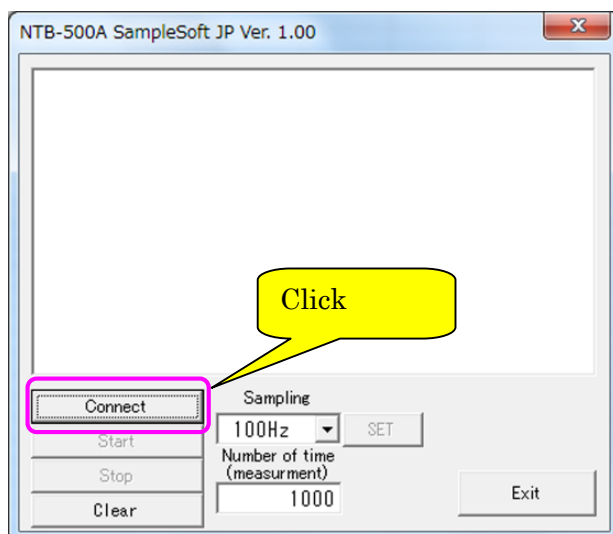


Click the Start button on the Sheet1.

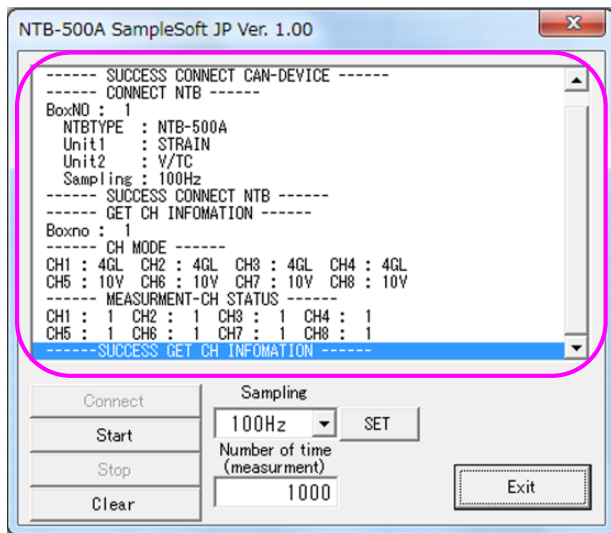


The window appears.

(3) Click the Connect button.



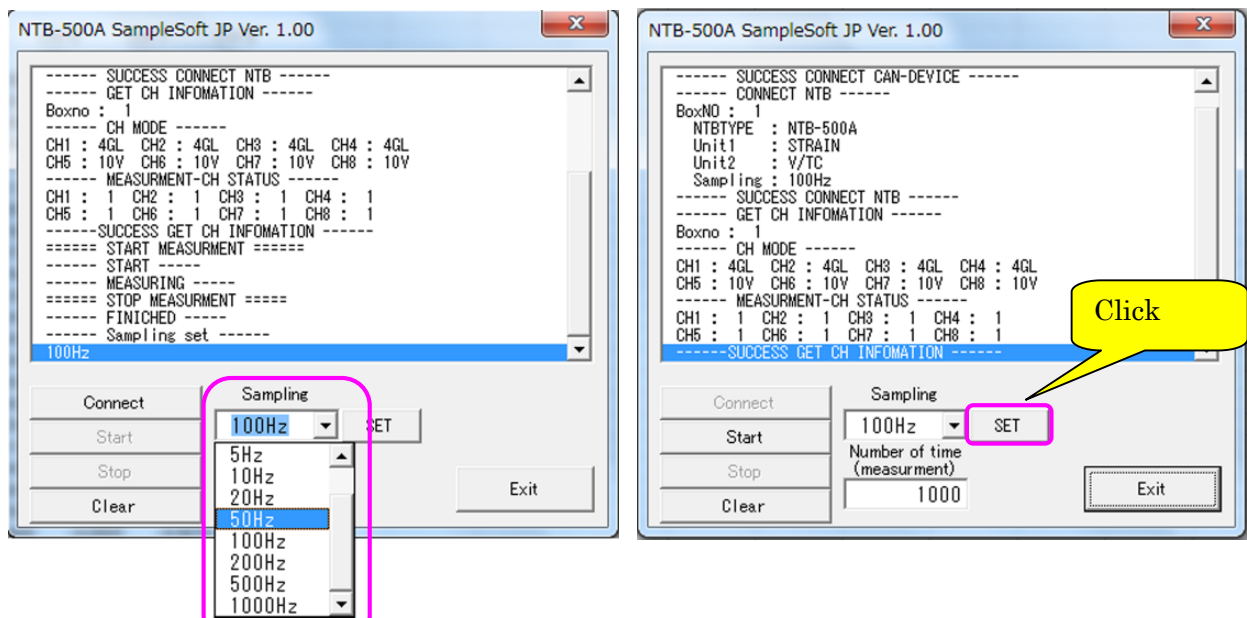
(4) After the connection succeeded, the information of the connected NTB-500A units appear.



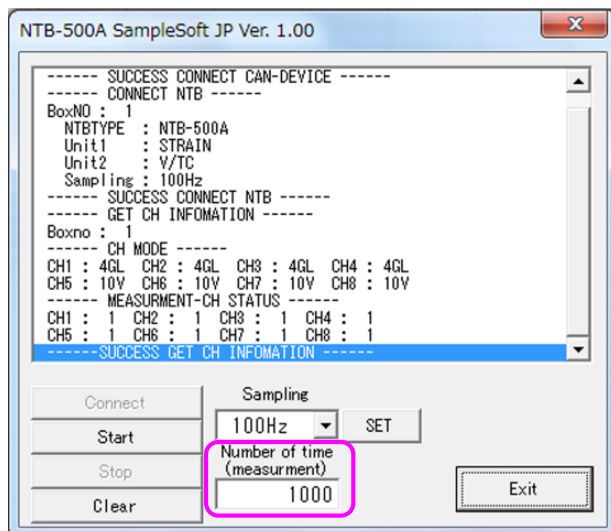
BoxNO : 1 BOX No.
 NTBTYPE : NTB-500A Model of the NTB
 Unit1 : STRAIN } Measuring unit
 Unit2 : V/TC
 Sampling : 100Hz Sampling frequency
 ----- SUCCESS CONNECT NTB -----
 ----- GET CH INFORMATION -----
 Boxno : 1
 ----- CH MODE -----
 CH1 : 4GL CH2 : 4GL CH3 : 4GL CH4 : 4GL } CH Mode
 CH5 : 10V CH6 : 10V CH7 : 10V CH8 : 10V
 ----- MEASUREMENT-CH STATUS -----
 CH1 : 1 CH2 : 1 CH3 : 1 CH4 : 1 } Measurement ON/OFF
 CH5 : 1 CH6 : 1 CH7 : 1 CH8 : 1 } OFF=0 ON=1

(5) Select the sampling frequency from the Sampling drop-down list.

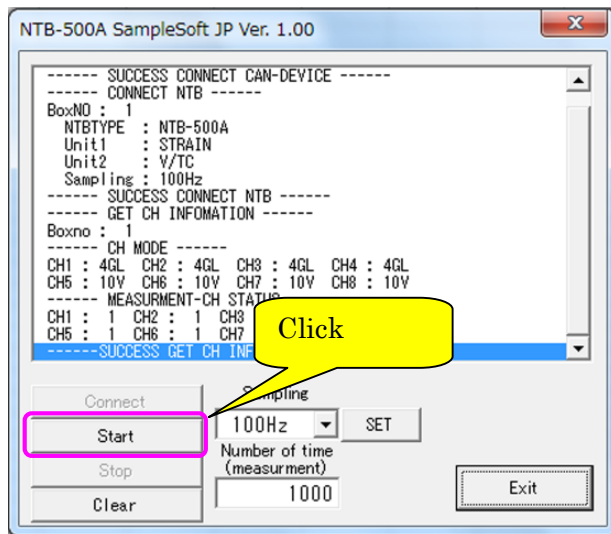
To set the pre-set sampling frequency of the NTB-500A, click the **SET** button.



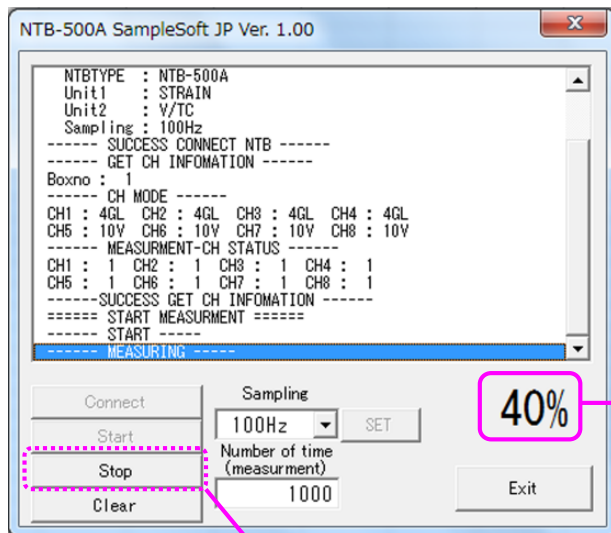
(6) Input the Number of time (measurement). (1 to 30,000)



(7) Click the **Start** button to start measurements.



During measurements

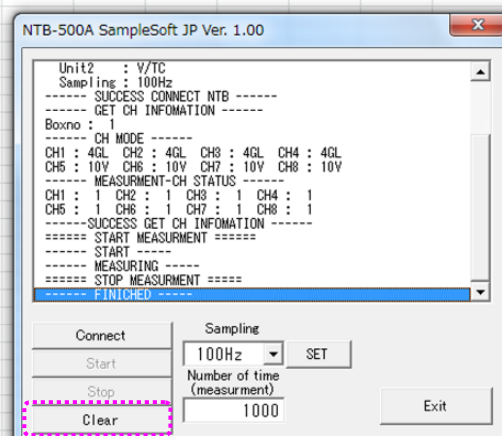


The rates shown are a percentage of the current status. Every measurement completes at 100%.

To stop the measurement before completing 100%, click the **Stop** button.

The measured data appears on the spreadsheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2																
3																
4																
5	CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7	CH 8								
6	4GL	4GL	4GL	4GL	10V	10V	10V	10V								
7	24474.5	-1.1	-1.4	160.2	3.7165999	3.7176001	3.7179999	3.7172								
8	24482.5	-1.4	-2.1	168.5	3.9156001	3.9171	3.9165001	3.9156001								
9	24467.301	-1.3	-2	155.10001	4.1012001	4.1020002	4.1027002	4.1019001								
10	24476.4	-0.7	-1.6	166.5	4.2688999	4.2701998	4.2698002	4.2690001								
11	24466.801	-1.3	-1.5	155.10001	4.4197998	4.4203	4.4211998	4.4204998								
12	24475.4	-1.4	-1.6	166.3	4.5535998	4.5546999	4.5545001	4.5538001								
13	24466.1	-0.9	-1.5	153.89999	4.6711001	4.6712999	4.6722999	4.6718998								
14	24475.1	-1.2	-1.8	166.7	4.7694001	4.7701001	4.7701001	4.7695999								
15	24465.1	-0.8	-2	154.39999	4.8502998	4.8502002	4.8513999	4.8511								
16	24474.199	-0.7	-1.6	165.5	4.9110999	4.9116001	4.9116998	4.9113998								
17	24464.5	-1.2	-1.6	153.7	4.9527998	4.9524999	4.9538002	4.9535999								
18	24473.801	-1.1	-1.9	166.5	4.9738998	4.9741001	4.9744	4.9745002								
19	24464.9	-0.8	-1.8	155.2	4.9777002	4.9770999	4.9787002	4.9791002								
20	24473.199	-1	-2	166.60001	4.9607	4.9607	4.9611001	4.9612999								
21	24463.801	-1.1	-1.9	156.5	4.9250998	4.9243002	4.9259	4.9264002								
22	24472.1	-0.8	-2.1	167	4.8695002	4.8691001	4.8698001	4.8702998								
23	24463.9	-0.9	-2.2	157.10001	4.7951002	4.7939	4.7958002	4.7964001								
24	24471.4	-1.2	-2.2	166.89999	4.6999002	4.6992998	4.7000999	4.7007998								
25	24463.699	-0.8	-2	156.89999	4.5893002	4.5879002	4.5889	4.5907998								
26	24470.4	-0.9	-1.7	167.39999	4.4597001	4.4589	4.4597998	4.4607								
27	24464.1	-1	-1.7	156.2	4.3130002	4.3112998	4.3134999	4.3146								
28	24470	-1	-1.6	166.39999	4.1501999	4.1490998	4.1501999	4.1513								
29	24463.199	-1.1	-1.8	156.7	3.9717	3.9697001	3.9721	3.9734001								
30	24470.5	-0.8	-1.4	166.8	3.7743001	3.7729001	3.7742	3.7755001								
31	24464.9	-0.8	-1.9	159.39999	3.5629001	3.5606999	3.5631001	3.5646999								
32	24470.4	-1.1	-2	169	3.3369999	3.3354001	3.3367	3.3383								
33	24464.1	-1.1	-1.6	159.89999	3.1006	3.0983	3.1006999	3.1024001								
34	24471.4	-0.9	-2	170.3	2.8506999	2.849	2.8503001	2.8520999								
35	24463.801	-0.6	-1.8	160.7	2.5913999	2.5889001	2.5913999	2.5933001								

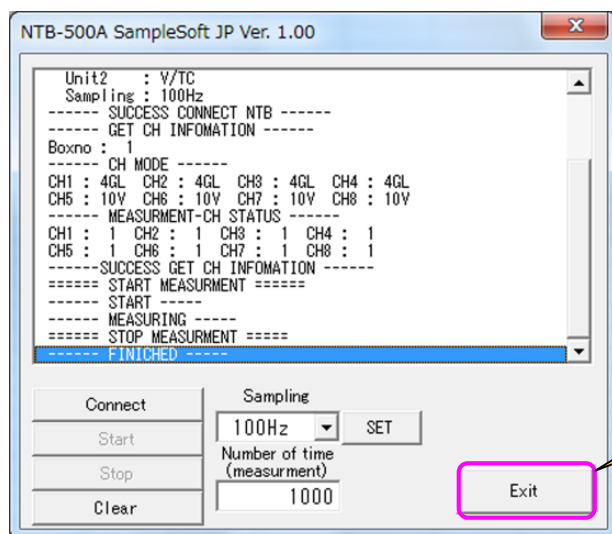


To clear the measured data from the cells, click the **Clear** button.

(8) End the simplified software.

Click the **Exit** button.

Be sure to click the **Exit** button to end the simplified software.



Click

(9) To save the measured data, save the file before closing the Microsoft Office EXCEL.